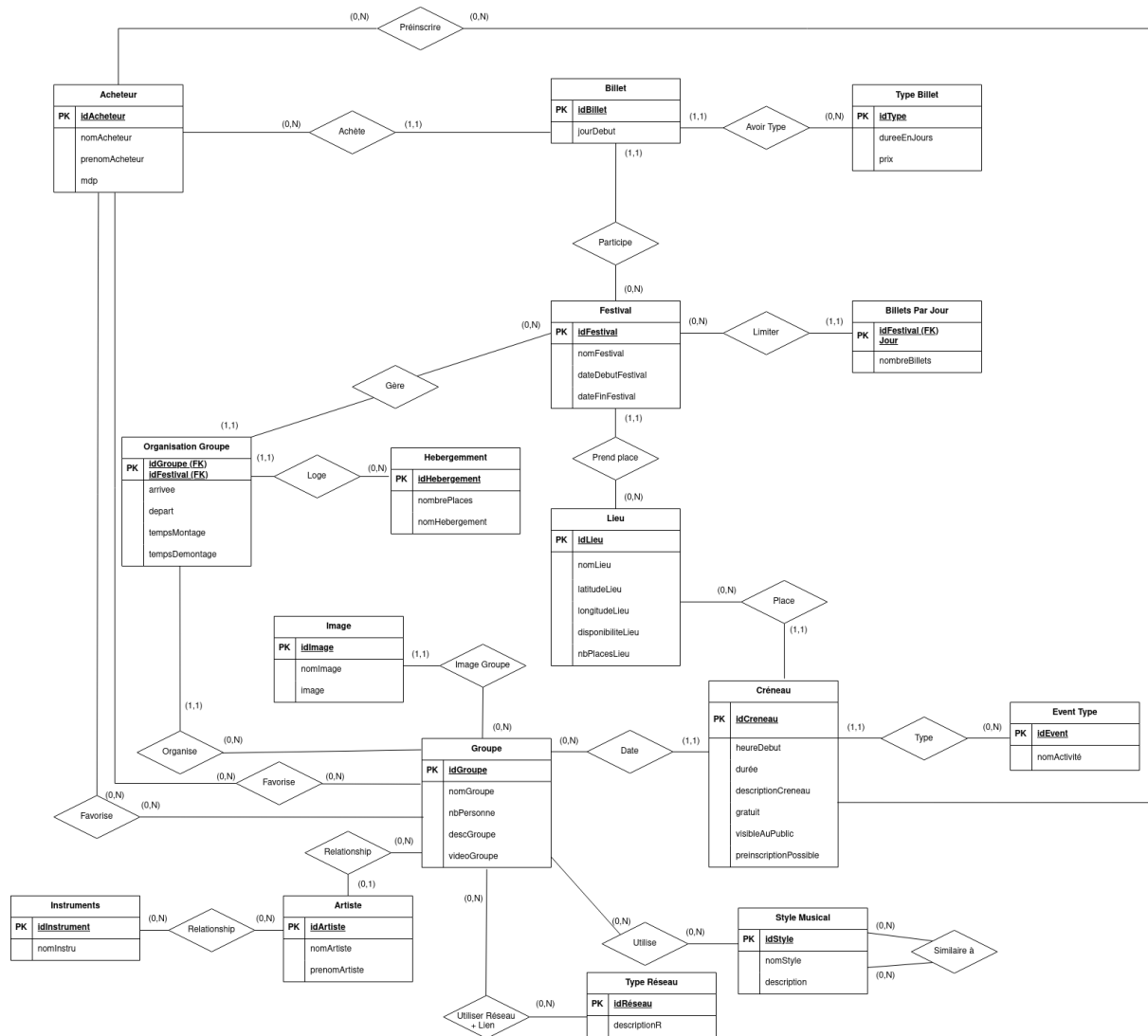


SAE - Fest-IUT'O

Ludmann Dorian, Lallier Anna, Pilet Colin et Lucidor Léo

I - Le schéma entité relation	2
II - Dépendance et contraintes	3
a) Dépendances fonctionnelles	3
1 - Billet	3
2 - Créneau	3
3 - Autres	3
b) Contraintes de la BD	3
1 - Les checks	3
2 - Les triggers	4

I - Le schéma entité relation



Ce MCD représente l'ensemble des dépendances fonctionnelles présentées par le sujet, et il cherche à rendre l'exemple plus concret, en permettant de gérer plusieurs festivals au lieu d'un seul.

II - Dépendance et contraintes

a) Dépendances fonctionnelles

1 - Billet

Hors les dépendances élémentaires, on retrouve des dépendances fonctionnelles à travers l'ensemble de la base de données.

Tout d'abord, un idTicket est relié (1,1) vers l'idAcheteur de quiconque le possède.

Ce même billet a ensuite un idType (1,1), qui permet de définir la durée du ticket (1J, 2J ou toute la durée du festival) ainsi que son prix.

Enfin ce billet permet de participer à un seul festival (1,1).

2 - Créneau

Afin qu'un créneau contienne l'ensemble des informations nécessaire, l'idCreneau est donc relié de manière unique (1, 1) vers idEvent pour définir quel type d'événement comprend le créneau, de même, il est relié à un idLieu (1, 1) pour définir le lieu de l'événement lors de ce créneau. Enfin, l'idCreneau est aussi relié à l'idGroupe (1, 1) pour définir quel groupe d'artiste est le sujet de l'événement.

3 - Autres

Il existe trois autres dépendances fonctionnelles qui servent de sécurité au bon fonctionnement de la base de données :

- Une instance de OrganisationGroupe est reliée de manière unique (1,1) à un festival ainsi qu'à un hébergement afin de certifier qu'un groupe puisse être hébergé. Cette dépendance permet aussi de vérifier que le groupe participe au bon festival.
- Une image appartient forcément à un groupe (1, 1), cette contrainte définit alors deux situations : une image ne peut pas ne pas être utilisée mais en conséquent, on sécurise le fait que l'image ne soit pas utilisée sur un groupe différent.
- Enfin, une instance de Billets Par Jour est reliée à un unique festival (1, 1) afin de certifier que ce nombre de billets s'applique au bon festival, ce qui permet d'éviter une situation où plus de billets seraient vendus que le nombre de billets déterminés pour un festival.

b) Contraintes de la BD

1 - Les checks

Tout d'abord, nous vérifions à l'aide d'un check que les heures de début et de fin du festival sont comprises entre 14 et 7h, puis on vérifie que la date de fin est supérieure à la date de début.

Dans la table billet par jour, un check a été implémenté afin de vérifier que le jour de cette

instance soit compris entre 1 et 31, afin de respecter la syntaxe de la date.

Pour ce qui est de l'hébergement, on vérifie que l'attribut nombreDePlaces ne soit pas nul.

Dans la table liaison musicale, qui permet de lier des styles musicaux similaires, on vérifie que le second idStyle ne soit pas égal à au premier afin d'éviter une liaison élémentaire.

Lors de l'ajout d'un groupe, on vérifie que l'attribut nbPersonnes ne soit pas nul, comme pour l'hébergement.

Quant à la table Organisation Groupe, on se doit de vérifier que la date d'arrivée du groupe soit inférieure à celle de leur départ, de la même façon qu'on le fait sur les dates du festival.

Enfin, pour l'utilisation de la table créneau, on vérifie alors que l'heure de début est comprise entre 14 et 4h du matin afin que le créneau soit compris pendant les horaires du festival.

2 - Les triggers

Pris par le temps, nous n'avons actuellement qu'un seul trigger de fonctionnel, mais nous savons déjà ce qu'il faut implémenter par la suite.

Le trigger implémenté fonctionne sur la classe LiaisonMusicale, et il vérifie que la liaison implémentée (X, Y) n'existe pas déjà dans la BD sous la forme (Y, X), afin d'éviter les doublons inutiles, par le fait que les relations ne sont pas orientés.

De ce qui est des autres triggers, voici une liste des triggers que nous souhaitons implémenter dans le futur :

- Vérifier que lors de l'insertion d'un créneau, son horaire de fin (soit horaire de début + durée) soit inférieur à 4h du matin. De plus, il faut vérifier que le début et la fin du créneau soient compris dans les dates du festival.
De plus, il faut aussi vérifier la disponibilité du lieu, ainsi que celle du groupe qui doit performer sur ce créneau.
- Il faut ensuite vérifier que lorsqu'un utilisateur se pré-inscrit à un créneau, ce créneau ait la pré-inscription active. Si c'est le cas, il faut vérifier que le nombre de préinscriptions à ce créneau ne dépasse pas la capacité du lieu concerné par le créneau.
- Enfin, le dernier trigger doit se faire lorsqu'un utilisateur achète un ticket. Celui-ci doit vérifier que le nombre de tickets vendus pour la journée n'est pas dépassé.
La difficulté de ce trigger est que les tickets peuvent être valides sur plusieurs jours. Pour faciliter son implémentation, une fonction retournant le nombre de tickets valides sur une date donnée en paramètre devra être créée.